## REMARKS/ARGUMENTS

These remarks are in response to the Final Office Action dated May 18, 2005. Claims 1-90 are pending. The Examiner rejected claims 1-5, 10-12, 14-17, 22-24, 26-31, 36-38, 40-43, 48-50, 52-58, 64-65, and 67-90 under 35 U.S.C. §102(e) as being anticipated by Drexter (U.S. App. No. 2002/0046248). Claims 6-9, 32-35 and 59-63 were rejected under 35 U.S.C. §103(a) as being unpatentable over Drexler in view of Demers et al. (U.S. Patent No. 5,870,761). Claims 13 and 39 were rejected under 35 U.S.C. §103(a) as being unpatentable over Drexler in view of Huth et al. (U.S. Patent No. 6,704,742). Claims 18-21, 25, 44-47, 51, and 66 were rejected under 35 U.S.C. §103(a) as being unpatentable over Drexler in view of Poskanzer (U.S. Patent No. 6,658,426).

In so doing, the Examiner stated:

> As to claim 1, Drexter teaches a method for converting messaging data into a relational table format in a database system, wherein the messaging data is within a messaging system (see page 1, paragraph 0002), the method comprising the steps of:
>
> (a) providing a table formatting specifications; (see page 2, paragraph 0029);
>
> (b) utilizing the plurality of table formatting specifications to automatically build and store a table function in the database system (see page 3, parapgraph 0034, where it is inherent that the associations (functions) are stored if they are going to be retrieved or recalled);
>
> (c) invoking the table function to access the messaging data (see pages 2-3, paragraphs 0030-0033); and
>
> (d) converting the messaging data by the table function into specific data types according to the plurality of table formatting specifications, wherein the messaging data is transformed into the relational table format (see page 3, paragraph 0033).
>
> . . .
>
> As to claim 67, Drexter teaches a system for generating a customized invocation mechanism (see page 1, paragraph 0002), comprising:
>
> an interface for receiving customizations (see page 3, paragraph 0034-0037); and
>
> a software module coupled to the interface for building an invocation mechanism based on the customization specifications and

> storing the invocation mechanism in a database (see page 3, paragraph 0034, where it is inherent that the associations (functions) are stored if they are going to be retrieved or recalled), wherein the invocation mechanism is invokable by the database for accessing data external to the database (see page 3, paragraphs 0036-0037).

The Examiner further asserts that:

> Drexler does disclose saving the function on the database system (see paragraph 0034, where it is inherent that if the association is going to be recalled at a later time it is saved). Drexter clearly discloses that one embodiment of the present invention is for it to be executed on a (single) computer system (see paragraph 0034). No where in Drexter is there evidence that the invention requires multiple systems or that parts of the invention would be executed on different systems. . . . It is noted that a database is simply a storage area, and that a database system is needed to invoke any program, application, or mechanism, and it is therefore assumed that the applicant is referring to the database system when reciting the limitation "wherein the invocation mechanism is invokable by the database" in claims 67, 75, and 83.

Applicants respectfully disagree.

The present invention is directed to building and using a table function that accesses messaging data in a messaging system and converts it into relational table format, i.e., a row with columns of desired data types. (Specification, page 9, lines 10-17; page 10, lines 10-11). According to the present invention, a user provides table formatting specifications that are utilized by an application to automatically build the table function. (Specification, page 10, lines 13-17). The table function is then *stored in the database*, along with user-defined functions (UDFs) and stored procedures (Specification, page 11, lines 19-23), where it can be invoked via an SQL statement. When invoked, the table function automatically retrieves messaging data stored in a message queue and converts that data into specific data types in relational table format (Specification, page 15, lines 6-10).

According to a version of the present invention, the table function can be a UDF, or some other statement *stored in the database* and *invocable by the database*. As such, the table function expands the functionality of existing databases by enabling the database to convert messaging data extracted from the message system into a relational table format automatically, without requiring additional middleware and/or application programs.

The present invention, as recited in claims 1 and 67, provides:

> 1. A method for converting messaging data into a relational table format in a database system, wherein the messaging data is within a messaging system, the method comprising the steps of:
>
> (a) providing a plurality of table formatting specifications;
>
> (b) utilizing the plurality of table formatting specifications to automatically build and store a table function in the database system;
>
> (c) invoking the table function from within the database system to access the messaging data; and
>
> (d) converting the messaging data by the table function into specific data types according to the plurality of table formatting specifications, wherein the messaging data is transformed into the relational table format.
>
> 67. A system for generating a customized invocation mechanism, comprising:
>
> an interface for receiving customizations; and
>
> a software module coupled to the interface for building an invocation mechanism based on the customization specifications and storing the invocation mechanism in a database, wherein the invocation mechanism is invokable by the database for accessing data external to the database.

Independent claims 27 and 53 are computer product and system claims, respectively, having scopes similar to that of claim 1. Independent claims 75 and 83 are method and computer product claims, respectively, having scopes similar to that of claim 67.

In contrast, Drexler is directed to an application program module that retrieves a message, parses the message into strings (when appropriate), and then applies a database association to the parsed strings in order to store the strings in fields in a database. (FIG. 2; ¶0029-¶0033). The database association is configured by the user (¶¶ 0034-0036; see FIG. 4) and stored in a file system in the computer system (¶0041). By checking a box, the user can indicate that messages pertaining to a particular database association be automatically retrieved periodically (¶0042).

Independent Claims 1, 27, 53, 67, 75 and 83.

Applicants respectfully submit that Drexler fails to teach or suggest the present invention, as recited in claims 1, 27, 53, 67, 75 and 83. In particular, Drexler does not teach or suggest storing "a table function in the database system," as recited in claims 1, 27 and 53, or "storing the invocation mechanism in a database," as recited in claims 67, 75 and 83. In the present invention, the table function and invocation mechanism is stored in the database or database system (the terms are interchangeable). This allows the table function and invocation mechanism to be invoked via an SQL statement understood by the database/database system. Accordingly, a separate application module is not required to use the table function and invocation mechanism.

In Drexler, FIG. 1 shows that the Email to Database Import Program 40, the database association 60, and the database 80 are separate components (¶¶0025-0027). There is no teaching or suggestion that the association 60 is stored in the database/database system, as recited in claims 1, 27, 53, 67, 75 and 83. In fact, Drexler explicitly states that the associations 60 can be found in "memory files, such as those on a floppy diskette, on the computer's hard drive, or a network hard drive." (¶0041).

In the Final Office Action, the Examiner states that the association is inherently stored because they can be used repeatedly. Applicants do not dispute this. The Examiner also states that the association is probably stored on the same computer system as the database/database system. Applicants do not dispute this either. Nevertheless, the Examiner seems to contend that the database/database system and the computer system are one and the same and that therefore, because the association is stored in the *computer* system, it is also stored in the database system. Applicants strongly disagree.

Contrary to the Examiner's position, a database *is not* "simply a storage area." It is well known in the industry and in the art that a "database" is defined as a set of related files that is created and managed by a database management system (DBMS). (See http://www.techweb.com/encyclopedia). According to another definition,

> [a] database is a collection of data elements (facts) stored in a computer in a systematic way, such that a computer program can consult it to answer questions. . . . The computer program used to manage and query a database is known as a database management system (DBMS). . . . Strictly speaking, the "database" is the collection of facts and the software is the "database management system" or DBMS. *However, in practice, many database administrators and programmers use the term "database" to cover both meanings.*

(See http://en.wikipedia.org/wiki/Database). Drexler explicitly states that the database 80 "may be a commercially available or privately created program . . . . The database 80 preferably includes a number of records, tables and/or fields to which data may be recorded." (¶0027). It is well known that while the database/database system can reside in a single computer system, such as a server, only the "collection of data elements" stored in the database/database system are *managed* by the database/database system. Other files, information and data stored in the file system of the computer system, outside of the database/database system, are managed by the operating system or some other application(s) in the computer system, and *not* by the database/database system.

Applicants respectfully submit that storing Drexler's associations in a memory file in the computer system or in a storage device of the computer system is not equivalent to storing the associations in the database/database system. Accordingly, Applicants respectfully submit that Drexler fails to teach or suggest storing "a table function in the database system," as recited in claims 1, 27 and 53, or "storing the invocation mechanism in a database," as recited in claims 67, 75 and 83.

In addition, Drexler also fails to teach or suggest "invoking the table function from within the database system," as recited in claims 1, 27 and 53, or an "invocation mechanism [that] is invokable by the database," as recited in claims 67, 75 and 83. In the present invention, the table function can be invoked from within the database/database system via an SQL statement. Because the database/database system understands the table function, it can invoke and execute it.

In Drexler, the Import Program 40 has access to the association 60 and to the database 80. According to Drexler, the "utility program 40 uses the association 60 to associate and save certain data from the email message 10 to appropriate records, tables or fields in the database 80. (¶ 0028). Thus, the utility program 40, and not the database, invokes the association. Indeed, Drexler's database is passive, e.g., the database only receives and stores data strings in fields. The database does not have the power to invoke or the ability to understand the association.

In the Final Office Action, the Examiner asserts that Drexler teaches "reusing the association (function) using the name given to the association (see paragraph 0034) which is read on invoking the application from within the database system." Again, the Examiner seems to contend that the database system and the computer system are one and the same. Based on the discussion above, Applicants respectfully submit that while the database system can reside in the computer system, the computer system and the database system are not one and the same. Thus, using a utility application program in the computer system to invoke the association is not equivalent to invoking the association "from within the database system," as recited in claims 1, 27 and 53, and having an association that is invokable by the application program is not equivalent to an "invocation mechanism [that] is invokable by the database," as recited in claims 67, 75 and 83.

Finally, Drexler fails to teach or suggest "*converting* the messaging data . . . into specific data types, *wherein the messaging data is transformed into the relational table format*," as recited in claims 1, 27 and 53. In the present invention, the table function *converts* data from the message into specific data types according to the plurality of table formatting specifications so that the data can be *transformed* into a relational table format. The relational table format is a row with columns of desired data types. (Specification, page 10, lines 10-11). In Drexler, the association corresponding to a message associates parsed data strings with database fields (¶0033). The association can perform other functions that manipulate the parsed data so that it is consistent with other data stored in the database (¶¶ 0067). Nevertheless, nothing teaches or suggests that the association transforms the messaging data "into the relational table format," as recited in claims 1, 27 and 53.

Based on the foregoing, Applicants respectfully submit that Drexler fails to teach or suggest the present invention, as recited in claims 1, 27, 53, 67, 75 and 83. Accordingly those claims are allowable over Drexler.

Dependent Claims in General

Claims 2-26, 28-52, 54-66, 68-75, 76-82 and 84-90 depend from claims 1, 27, 53, 67, 75 and 83, respectively. Thus, claims 2-26, 28-52, 54-66, 68-75, 76-82 and 84-90 are also allowable over Drexler. Because the secondary references fail to remedy the deficiencies of Drexler, Applicants respectfully submit that dependent claims 2-26, 28-52, 54-66, 68-75, 76-82 and 84-90 are allowable over Drexler in view of the cited references.

Dependent Claims 2, 28 and 54

Applicants respectfully submit that claims 2, 28 and 54 are allowable over Drexler because they depend from claims 1, 27 and 53, respectively, and because Drexler fails to teach or suggest a table function that "invokes at least one messaging function within the database system," as recited in claims 2, 28 and 54. In the present invention, the table function and the messaging function are UDFs that are stored in the database system. The table function invokes the messaging function from within the database system to retrieve the messaging data.

In Drexler, the import utility program receives the email message and uses the association to associate parsed data from the email message to appropriate records, tables or fields in the database. (¶0028). Nothing in Drexler teaches or suggests that the association invokes the utility program from within the database system, as recited in claim 2. In the Final Office Action, the Examiner asserts that Drexler teaches this feature at ¶0042. That paragraph states that the user who is configuring the association can "enable automated data import by the association" by checking a box. By checking the box, the association is scheduled for an automated email to database import process, which means the utility import program will receive email periodically and check to see if the received email should be processed by the association. Nothing in paragraph 0042 teaches or suggests that the association "invokes at least one messaging function within the database system," as recited in claims 2, 28 and 54.

Based on the above reasoning, Applicants respectfully submit that Drexler fails to teach or suggest the present invention, as recited in claims 2, 28 and 54.

Dependent Claims 3, 29 and 55

Applicants respectfully submit that claims 3, 29 and 55 are allowable over Drexler because they depend from claims 1, 27 and 53, respectively, and because Drexler fails to teach or suggest that the table function and the messaging function "are user-defined functions within the database system," as recited in claims 3, 29 and 55. As stated above, the table function and the messaging functions are user-defined functions (UDFs) within the database system. UDFs are well known in the art. In general, a UDF is a routine that has been defined or programmed by the user of the system and has been included in a standard library of functions. In the context of a database system, the UDF is a set of SQL statements with an assigned name that is stored in the database so that it can be invoked by other UDFs or within an SQL statement. Nothing in Drexler teaches or suggests that the association or the import utility are "are user-defined functions within the database system," as recited in claims 3, 29 and 55.

In the Final Office Action, the Examiner asserts that Drexler teaches this at ¶0034. That paragraph, however, merely discusses FIG. 3 and how a user creates an association. Also the association is created by the user, it is not a "user-defined function within the database system." Also, there is not teaching or suggest in paragraph 0034 that the messaging function, e.g., the import utility program, is also a UDF within the database system. Accordingly, Applicants respectfully submit that Drexler fails to teach or suggest the present invention, as recited in claims 3, 29 and 55.

## Conclusion

In view of the foregoing, Applicants submit that claims 1-90 are allowable over the cited references. Applicants respectfully request reconsideration and allowance of the claims as now presented.

Applicant's attorney believes that this application is in condition for allowance. Should any unresolved issues remain, Examiner is invited to call Applicant's attorney at the telephone number indicated below.

Respectfully submitted,
SAWYER LAW GROUP LLP

July 14, 2005
Date

Joyce Tom
Attorney for Applicant
Reg. No. 48,681
(650) 493-4540